

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky a výpočetní techniky

Absolvování individuální odborné praxe
Individual Professional Practice
in the Company

Zadání bakalářské práce

Student:

Aneta Opletalová

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Absolvování individuální odborné praxe
Individual Professional Practice in the Company

Jazyk vypracování:

čeština

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: KVADOS, a.s.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c) Zvolený postup řešení zadaných úkolů.
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Peter Chovanec, Ph.D.**

Konzultant bakalářské práce: Bc. Lukáš Sadovský

Datum zadání: 01.09.2018

Datum odevzdání: 30.04.2019




doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry


prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

V Ostravě dne 23. 4. 2019



Aneta Opletalová

Kvados, a. s.

Novoveská 1139/22

Ostrava – Mariánské Hory

Souhlas se zveřejněním bakalářské práce

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Ostravě dne 15. 4. 2019


.....

Bc. Lukáš Sadovský

Abstrakt

Tato bakalářská práce vychází z mé praxe u firmy Kvados, a. s.. Společnost vytváří software pro řízení obchodu, logistiky a vnitrofiremních procesů, který je uplatňován po celém světě. Práce popisuje vývojové metody a technologie, se kterými jsem se v průběhu praxe setkala, a můj podíl při vývoji webové aplikace myTeam, která slouží pro zautomatizování vnitrofiremních procesů.

Klíčová slova: Kvados, a. s., C#, SQL, Angular, TFS, HTML5, TypeScript

Abstract

This bachelor thesis is based on my professional practice at Kvados, a. s. The company creates software solutions that focus mainly on project management and business and logistics processes. The bachelor thesis describes development methods and technologies which I have been using during the professional practice, and my contribution to development of web application myTeam, which helps with automation of company's internal processes.

Key Words: Kvados, a. s., C#, SQL, Angular, TFS, HTML5, TypeScript

Obsah

Seznam použitých symbolů a zkratek.....	7
Seznam ilustrací a tabulek.....	8
Seznam zdrojových kódů.....	9
1 Úvod.....	10
1.1 Kvados, a. s.	10
1.2 myTeam.....	10
2 Technologie	11
2.1 TFS.....	11
2.2 C#	12
2.3 Angular.....	12
2.4 SSMS.....	12
3 myTeam.....	13
3.1 Uživatelské rozhraní.....	13
3.2 Implementace	15
4 Praxe.....	16
4.1 Agilní vývoj – scrum.....	16
4.2 Postup plnění úkolů	17
4.3 Modul Směrnice	18
4.4 Modul Úkoly	27
4.5 Modul Objednávky.....	29
4.6 Modul Podatelna.....	32
4.7 Sjednocení e-mailu	33
4.8 Issues	34
5 Závěr.....	35
Literatura.....	36

Seznam použitých symbolů a zkratek

CSS	Cascading Style Sheets
DMS	Document Management System
HTML	HyperText Markup Language
SQL	Structured Query Language
SSMS	SQL Server Management Studio
TFS	Team Foundation Server
TS	TypeScript
VS	Visual Studio
VSCode	Visual Studio Code

Seznam ilustrací a tabulek

1 TFS - Detail changesetu uloženého na serveru.....	11
2 SQL Server Profiler – zachycení provozu	12
3 Modul Směrnice	14
4 Struktura modulu Směrnice - serverová část	15
5 Scrum board.....	16
6 RightsViewer - základní role a jejich práva k vybraným úkonům	17
7 Schvalovací proces směrnice – přehled stavů.....	18
8 Panel Schválit	19
9 Panel Publikovat.....	20
10 Panel Kontrola platnosti směrnice	22
11 Panel Určeno pro a Odstranit záznam	25
12 Zvoneček s notifikací.....	27
13 Modul Úkoly s pohledem pro směrnice.....	28
14 Objednávky vydané – Panel Nový záznam	29
15 Odeslaná zásilka – Panel Nový záznam	32
16 E-mail Zamítnutí objednávky	34
17 E-mail Pozvánka na poradou	34

Seznam zdrojových kódů

1 Relace mezi panely	23
2 Extenze pro přenos dat na server	23
3 Validace povinných polí	24
4 Vygenerování úkolu	24
5 Periodická kontrola směrnice	26
6 Textbox Provozní jednotka	30
7 Skládání SQL dotazu pro rozbalovací menu Provozní jednotky	30
8 Triggery pro spuštění extenze	30
9 Přednastavení pole Provozní jednotka – část 1	31
10 Přednastavení pole Provozní jednotka – část 2	31
11 Sekce Dopis Online – podmínka zobrazení	33

1 Úvod

Tato bakalářská práce vychází z mé praxe u firmy Kvados, a. s. [1], kde pracuji jako software developer. K této pozici jsem se dostala skrze kariérní veletrh Kariéra PLUS, který se každoročně pořádá na VŠB – TUO. Pro bakalářskou praxi jsem se rozhodla především z toho důvodu, že jsem chtěla získat zkušenosti z vývoje reálného produktu, které obohatí mé dosud nabyté teoretické znalosti.

V následujících kapitolách tedy budu popisovat vybrané úkoly, kterým jsem se v průběhu praxe věnovala.

1.1 Kvados, a. s.

Společnost Kvados, a. s., založena roku 1992, se zabývá vývojem vlastních softwarových řešení a poradenstvím. Produkty se orientují především na správu a podporu firemních procesů, například v podobě komplexních ERP systémů, ale nabízí také další produkty, které mají užší zaměření na konkrétní problematiku jako je logistika či podpora prodeje.

1.2 myTeam

V rámci své praxe jsem byla přiřazena do týmu, který se věnuje vývoji webové aplikace myTeam [2]. Je to modulární portálové řešení pro sdílení informací, úkolů, řízení procesů a jejich automatizaci. Pokrývá širokou škálu vnitrofiremních procesů, přičemž každý z nich má svůj vlastní oddíl – modul.

Jedním ze základních modulů je centrální úkolovník, který poskytuje nástroj pro práci s úkoly a jejich připomínkování. Mezi další pokryté agendy patří podatelna, která zprostředkovává oběh korespondence a zásilek, nebo evidence objednávek a nákupů. Jednou z důležitých oblastí je také práce s dokumenty, fakturami, směrnicemi a smlouvami, a to včetně dohledu nad jejich schvalovacími procesy. Dále aplikace umožňuje organizaci porad a schůzek a správu vozového parku či nemovitostí. Další modul obsahuje evidenci pracovní doby spolu s žádankami o dovolenou.

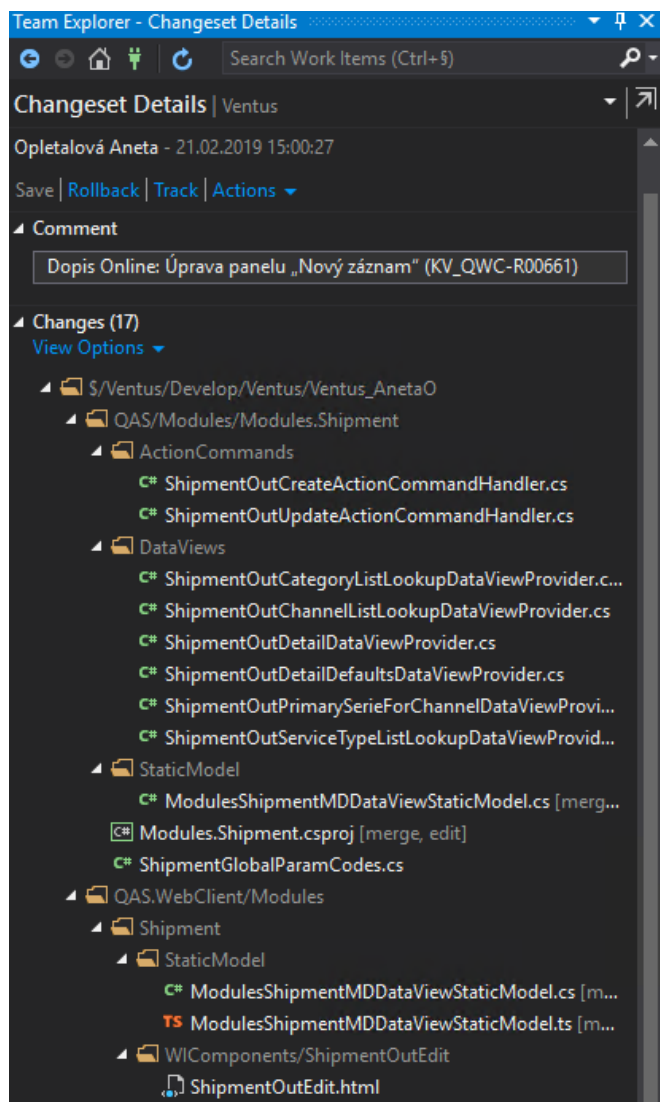
2 Technologie

V začátcích praxe jsem se musela seznámit s konkrétními nástroji a technologiemi, které firma ve své aplikaci používá. Mezi základní využívané technologie patří:

2.1 TFS

Team Foundation Server [3, 4] je produkt Microsoftu [5], který napomáhá spravovat životní cyklus aplikace po celou dobu jejího vývoje. Je možné jej používat společně s různými vývojovými prostředími, v mém případě to bylo Visual Studio [6]. Nástroj umožňuje mimo jiné sledovat a kontrolovat různé verze kódu (verzování), udržovat přehled nad požadavky, testování nebo využívat automatické buildy.

Při své praxi jsem si především osvojila praktiky verzování. TFS využívá metody slučování verzí, při níž mohou dva vývojáři současně upravovat stejný soubor a následně své změny nahrát do repozitáře na server. Všechny své změny (changeset) pak programátor nahrává (check-in) jako novou verzi na server. Na obrázku 1 je zobrazen detail jednoho changesetu s mými úpravami, který jsem nahrávala do repozitáře.



Obrázek 1: TFS - Detail changesetu uloženého na serveru

2.2 C#

C# [7] je vysokoúrovňový objektově orientovaný programovací jazyk založený na syntaxi jazyka C. Byl vyvinut společností Microsoft v roce 2000. Oproti C++ nepodporuje několikanásobnou dědičnost, avšak jedna třída může implementovat několik rozhraní. Jednou z výhod je například zjednodušení práce s proměnnými, které je možné zaobalit spolu s metodami get a set do property, které usnadňují správu proměnné. Zavádí také delegáty a eventy a o práci s pamětí se stará garbage collector.

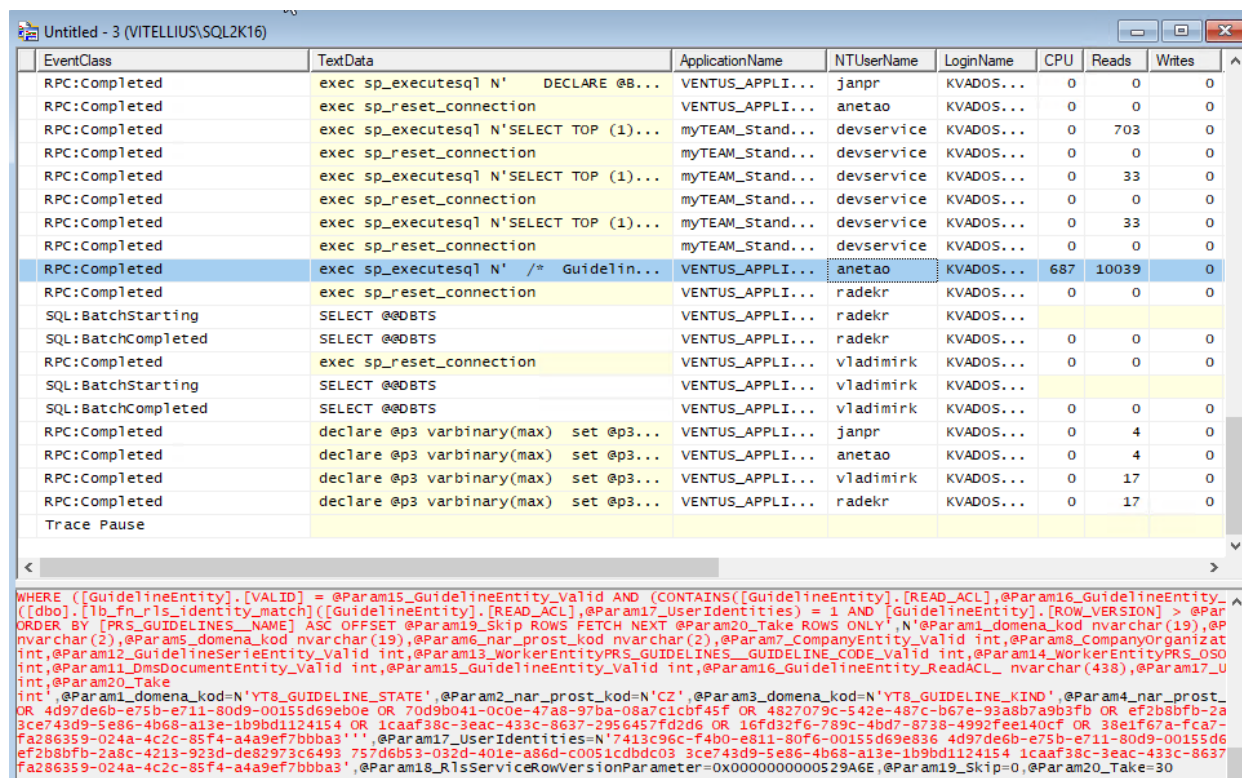
2.3 Angular

Angular [8] je open-source javascriptový webový framework, který se zaměřuje na tvorbu aplikací. Funguje na principu data-bindingu, kdy jsou do HTML kódu vloženy formátovací značky, které určují, jaké operace či data mají být na dané místo vloženy. Doplnuje tedy statický HTML kód o dynamický obsah. Hlavní výhodou je oddělenost aplikační logiky od logiky zobrazovací.

S touto technologií jsem se setkala úplně poprvé, ale díky školení, které nám bylo zprostředkováno, jsem se mohla v této oblasti rychleji zorientovat.

2.4 SSMS

SQL Server Management Studio [9] je nástroj, který zajišťuje komunikaci s Microsoft SQL Serverem [10]. Umožňuje přístup, konfiguraci a administraci SQL Serveru pomocí přehledného prostředí. Nejčastěji jsem SSMS využívala pro přístup k databázi a sledování provozu pomocí SQL Server Profileru [13], jehož ukázka je zachycena na obrázku 2.



EventClass	TextData	ApplicationName	NTUserName	LoginName	CPU	Reads	Writes
RPC:Completed	exec sp_executesql N' DECLARE @B...	VENTUS_APPLI...	janpr	KVADOS...	0	0	0
RPC:Completed	exec sp_reset_connection	VENTUS_APPLI...	anetao	KVADOS...	0	0	0
RPC:Completed	exec sp_executesql N'SELECT TOP (1)...	myTEAM_Stand...	devservice	KVADOS...	0	703	0
RPC:Completed	exec sp_reset_connection	myTEAM_Stand...	devservice	KVADOS...	0	0	0
RPC:Completed	exec sp_executesql N'SELECT TOP (1)...	myTEAM_Stand...	devservice	KVADOS...	0	33	0
RPC:Completed	exec sp_reset_connection	myTEAM_Stand...	devservice	KVADOS...	0	0	0
RPC:Completed	exec sp_executesql N'SELECT TOP (1)...	myTEAM_Stand...	devservice	KVADOS...	0	33	0
RPC:Completed	exec sp_reset_connection	myTEAM_Stand...	devservice	KVADOS...	0	0	0
RPC:Completed	exec sp_executesql N' /* Guidelin...	VENTUS_APPLI...	anetao	KVADOS...	687	10039	0
RPC:Completed	exec sp_reset_connection	VENTUS_APPLI...	radekr	KVADOS...	0	0	0
SQL:BatchStarting	SELECT @@DBTS	VENTUS_APPLI...	radekr	KVADOS...			
SQL:BatchCompleted	SELECT @@DBTS	VENTUS_APPLI...	radekr	KVADOS...	0	0	0
RPC:Completed	exec sp_reset_connection	VENTUS_APPLI...	vladimirk	KVADOS...	0	0	0
SQL:BatchStarting	SELECT @@DBTS	VENTUS_APPLI...	vladimirk	KVADOS...			
SQL:BatchCompleted	SELECT @@DBTS	VENTUS_APPLI...	vladimirk	KVADOS...	0	0	0
RPC:Completed	declare @p3 varbinary(max) set @p3...	VENTUS_APPLI...	janpr	KVADOS...	0	4	0
RPC:Completed	declare @p3 varbinary(max) set @p3...	VENTUS_APPLI...	anetao	KVADOS...	0	4	0
RPC:Completed	declare @p3 varbinary(max) set @p3...	VENTUS_APPLI...	vladimirk	KVADOS...	0	17	0
RPC:Completed	declare @p3 varbinary(max) set @p3...	VENTUS_APPLI...	radekr	KVADOS...	0	17	0
Trace Pause							

WHERE ([GuidelineEntity].[VALID] = @Param15_GuidelineEntity_Valid AND (CONTAINS([GuidelineEntity].[READ_ACL],@Param16_GuidelineEntity_([dbo].[fn_fn_rls_identity_match]([GuidelineEntity].[READ_ACL],@Param17_UserIdentities) = 1 AND [GuidelineEntity].[ROW_VERSION] > @Par ORDER BY [PRS_GUIDELINES].[NAME] ASC OFFSET @Param19_Skip ROWS FETCH NEXT @Param20_Take ROWS ONLY' N'@Param1_domena_kod nvarchar(19),@P nvarchar(2),@Param5_domena_kod nvarchar(19),@Param6_nar_prost_kod nvarchar(2),@Param7_CompanyEntity_Valid int,@Param8_CompanyOrganizat int,@Param12_GuidelineSerieEntity_Valid int,@Param13_WorkerEntityPRS_GUIDELINES_GUIDELINE_CODE_Valid int,@Param14_WorkerEntityPRS_OS0 int,@Param11_DmsDocumentEntity_Valid int,@Param15_GuidelineEntity_Valid int,@Param16_GuidelineEntity_ReadACL nvarchar(438),@Param17_U int,@Param20_Take int',@Param1_domena_kod=N'YT8_GUIDELINE_STATE',@Param2_nar_prost_kod=N'C2',@Param3_domena_kod=N'YT8_GUIDELINE_KIND',@Param4_nar_prost_ OR 4d97de6b-e75b-e711-80d9-00155d69eb0e OR 70d9b041-0c0e-47a8-97ba-08a7c1cbf45f OR 4827079c-542e-487c-b67e-93a8b7a9b3fb OR ef2b8bfb-2a 3ce743d9-5e86-4b68-a13e-1b9bd1124154 OR 1caaf38c-3eac-433c-8637-2956457fd2d6 OR 16fd32f6-789c-4bd7-8738-4992fee140cf OR 38e1f67a-fca7- fa286359-024a-4c2c-85f4-a4a9ef7bbba3' ,@Param17_UserIdentities=N'7413c96c-f4b0-e811-80f6-00155d69e836 4d97de6b-e75b-e711-80d9-00155d6 ef2b8bfb-2a8c-4213-923d-de82973c6493 757d6b53-032d-401e-a86d-c0051cddbdc03 3ce743d9-5e86-4b68-a13e-1b9bd1124154 1caaf38c-3eac-433c-8637 fa286359-024a-4c2c-85f4-a4a9ef7bbba3' ,@Param18_RlsServiceRowVersionParameter=0x0000000000529A6E,@Param19_Skip=0,@Param20_Take=30

Obrázek 2: SQL Server Profiler – zachycení provozu

3 myTeam

3.1 Uživatelské rozhraní

Úvodní stránka aplikace myTeam se skládá z levého bočního menu se všemi dostupnými agendami, horní lišty a nástěnky, kterou si uživatel může upravit podle vlastních potřeb. Po kliknutí na libovolnou agendu z hlavního menu se zobrazí základní panel s možnostmi pohledů a panel obsahující seznam položek, spadající pod danou oblast. Na obrázku 3 lze vidět tato základní sestava pro agendu týkající se směrnic.

Na tento seznam je možné aplikovat filtry, které pokrývají téměř každou vlastnost, která je u položky evidována, a to včetně časových údajů či zainteresovaných osob.

Po kliknutí na vybraný prvek seznamu se objeví další panel, ve kterém jsou zobrazeny jeho možnosti. Tento panel je vždy řízen přístupovými právy, tedy ne každý uživatel může vidět a provádět všechny dostupné akce. Základem ale vždy je zobrazení *Rychlého přehledu* s nejdůležitějšími informacemi a *Detailu* s podrobným souhrnem vlastností. Další volby jsou pak viditelné jen oprávněným osobám. Ty většinou zahrnují úkoly, které se k položce vztahují, a uživatel je má splnit, či nad nimi dohlížet.

Jednotliví pracovníci – uživatelé musí být v aplikaci rozřazeni do menších celků – organizačních jednotek, čímž získávají práva náležející jejich zařazení. Další práva jim jsou pak přidělena pomocí rolí, nebo, jak již bylo řečeno, při odpovědnosti za plnění nějakého úkolu.

Výše zmíněné úkoly jsou jedním z hlavních pilířů aplikace. Každý firemní proces takového úkoly generuje a myTeam byl vytvořen především pro jejich předávání a připomínkování. Na procesu se tak většinou podílí větší množství osob, ať už jde o žadatele, zpracovatele či schvalovatele, kteří postupně plní své povinnosti.

Dalším zásadním prvkem je DMS, tedy Systém pro správu dokumentů. Využívá se především v agendách týkajících se směrnic, smluv a faktur.

myTEAM® NG

Pokedy > Směrnice > Směrnice č. 00055 >

Směrnice

+

↺

↻

📄

☰

Nový

Obnovit

Filtr

Skupiny

Možnosti

Úkoly

Rizikový panel

Objednávky vydané

Faktury přijaté

Katalog sortimentu

Projekty

Organizace

Pracovníci

Firmy ve skupině

Administrace

Směrnice

Nemovitosti

Podatelna

Evidence prac. doby

směrnice

STAV

PLATNÁ

ČÍSLO

NÁZEV

SMĚRNICE

100 - výrobní oddělení

00055

Firma A, s.r.o.

směrnice BOZP

směrnice

08.11.2018^h

09.11.2018^h

0.2

0

✓

00011

Firma A, s.r.o.

Směrnice nova

100 - výrobní oddělení

Směrnice

15.10.2018^h

Neurčito

0.1

0

✓

00174

Firma A, s.r.o.

Směrnice o dledech

200 - Projektová kancelář, top od...

Směrnice

25.11.2018^h

Neurčito

2.3

0

✓

00107

Firma A, s.r.o.

Směrnice o firemních vozidlech

200 - Projektová kancelář, top od...

Směrnice

14.11.2018^h

Neurčito

2.0

0

00164

Firma A, s.r.o.

Směrnice o chování

200 - Projektová kancelář, top od...

Směrnice

25.11.2018^h

Neurčito

0.1

0

00154

Firma A, s.r.o.

Směrnice o pracovní kázní

200 - Projektová kancelář, top od...

Směrnice

25.11.2018^h

Neurčito

0.2

0

✓

00084

Firma A, s.r.o.

Směrnice o provozu

200 - Projektová kancelář, top od...

Směrnice

17.11.2018^h

Neurčito

2.0

0

00036

Firma A, s.r.o.

směrnice pokyny

100 - výrobní oddělení

Pokyny

13.11.2018^h

Neurčito

0.1

0

Obrázek 3 : Modul Směrnice

14

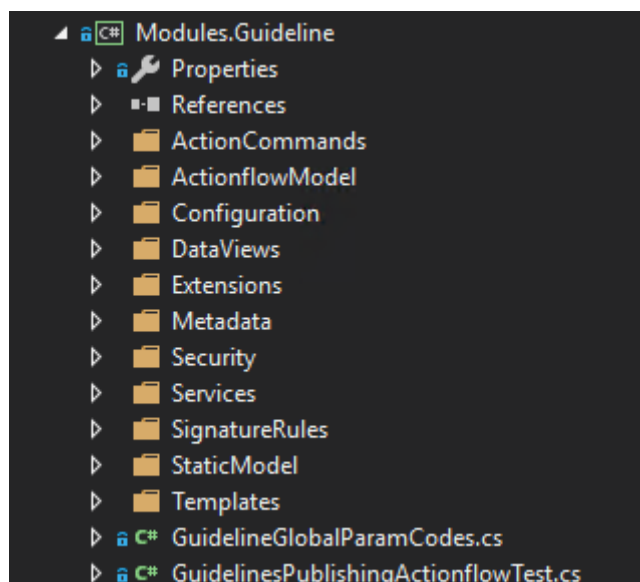
3.2 Implementace

Celá aplikace je rozdělena na dva projekty, kdy jeden obsahuje klientskou část – co a jak zobrazit, a druhá serverovou část – logiku. Používá se také aplikace VS Code, ve které se pro zjednodušení celého vývoje pracuje se všemi .html, .css a .ts soubory.

V projektu s klientskou částí se nachází především .xml soubory. Každý takový soubor většinou odpovídá jednomu panelu v aplikaci – tzv. *WorkItem*. *WorkItem* má svou danou strukturu a musí obsahovat konkrétní části, které zajišťují funkčnost a vzhled panelu. Mezi základní prvky patří seznam relací, což jsou vztahy mezi jednotlivými panely. Tyto vztahy jsou blíže určeny *Connectory*, které stanovují, jaká data se mezi panely přenáší. Typicky jde o různá identifikační čísla vztahující se k záznamu. Poslední důležitou součástí celého *WorkItemu* jsou *extensiony*. Jsou to typescript soubory, které ovládají obecné chování a zajišťují správný přenos dat. Dále je nutné poskládat samotný vzhled panelu pomocí bloků a komponent jako jsou formuláře, tlačítka, ikony nebo jednoduché popisky. Vše se pak generuje v podobě HTML, CSS a TS souborů do VS Code. Stisknutí tlačítek vyvolává spuštění skupiny metod na serveru – *ActionCommandu*. Samotná data se pak zobrazují pomocí volání serverových metod nazvaných *Dataview*,

Základem serverového projektu jsou, jak již bylo zmíněno výše, *ActionCommand* a *DataView*. Ty musí být přiřazeny pod správný jmenný prostor podle příslušné agendy. *ActionCommand* má také svou základní strukturu a dané metody, které musí být implementovány. Mezi ně patří metody validující vstupy z formulářů a funkce pro práci s daty, například jejich ukládání dat do databáze. *DataView* naopak data z databáze získávají prostřednictvím SQL dotazů, které se následně zpracují a posílají do klientské části v podobě objektu nazvaného *DataBag*. Jelikož se v celé aplikaci také pracuje s rolemi a právy uživatelů, další třídy mají na starost tato oprávnění zohledňovat a zpracovávat. Práva jsou vždy řešena na úrovni jednotlivých záznamů.

Na obrázku 4 je zobrazena struktura modulu Směrnice a rozřazení jednotlivých tříd.



Obrázek 4: Struktura modulu Směrnice - serverová část

4 Praxe

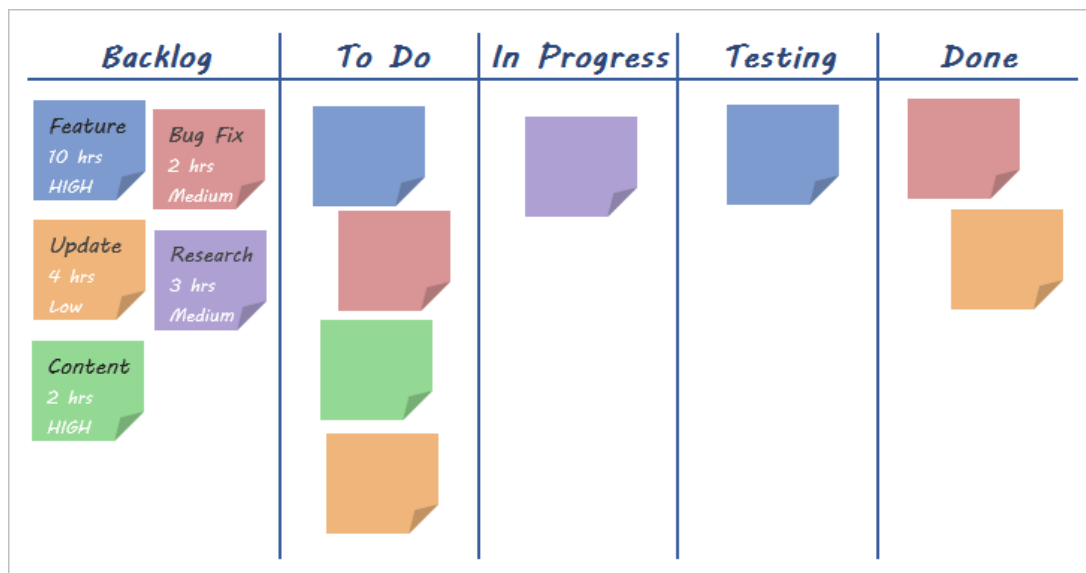
Náplní mé práce bylo zejména rozšiřování aplikace myTeam – přidávání nových panelů, funkcionalit nebo úpravy a vylepšování stávajících prvků. Celý vývoj aplikace byl založen na principech metodologie scrum.

4.1 Agilní vývoj – scrum

Scrum [12] je iterativní a inkrementální metodologie agilního vývoje softwaru. Oproti klasickému sekvenčnímu přístupu se nezaměřuje na striktní dodržování a dokončování fází životního cyklu softwaru. Naopak se soustředí na maximální schopnost týmu spolupracovat a reagovat na změny a nové požadavky.

Naší týmovou rutinou byla každodenní krátká setkání – daily standup, kdy jsme vždy ostatní kolegy seznámili s prací z předchozího dne, popř. s problémy, na které jsme narazili a je potřeba je řešit v rámci celého týmu. Potom jsme také nastínili plán na stávající den. Tato setkání vždy řídil tzv. scrum master, který musel mít přehled nad plněním aktuálního plánu.

Základní jednotkou vývoje softwaru je časově omezený úsek, v našem případě dvoutýdenní, který se nazývá sprint. Na začátku sprintu je vždy zorganizován sprint planning, na němž se setkává celý tým a plánuje práci, která se v daném sprintu má dokončit. Na nástěnce (viz obrázek 5), tzv. scrum boardu [11], jsme si pak vždy mohli rozebrat jednotlivé požadavky a přiřadit si je ke svému jménu do fáze „V řešení“. Následně po dokončení tohoto úkolu byl přesunut do fáze „V testování“, později pak „Hotovo“, případně „Oprava“.



Obrázek 5: Scrum board

4.2 Postup plnění úkolů

Pokaždé, když mi byl přidělen nový úkol, jsem si jej musela nejprve vyhledat v analýze, podle níž jsem pak zadání implementovala. O psaní podrobných analýz se starali analytici. Každý z nich měl na starost několik agend, kterým velmi dobře rozuměl. V analýze byla vždy podrobně popsána data, se kterými se má pracovat a jaké změny s nimi provádět. Pokud se jednalo o nějaký úkon, který byl ovlivněn právy uživatele nebo i stavem záznamu, bylo potřeba se ještě podívat do tabulky s přehledem práv. V této tabulce jsem si vyhledala svou zadanou funkci (*ActionCommand*) a v přehledu jsem zjistila, v jakém stavu se musí záznam nacházet, aby bylo možné tento úkon provést. Dále pak také jakou roli musí mít uživatel přiřazenou, aby měl tuto možnost zobrazenou a mohl úkon vykonat.

Tyto údaje jsem poté zapracovala do interního systému s názvem *RightsViewer*, skrze který se veškerá omezení následně aplikovala do myTeamu. Na obrázku 6 je vyobrazeno několik úkonů týkajících se zpracovávání směrnic. Ve sloupcích jsou základní práva, která se přiřazují uživatelům. Pokud chce uživatel vykonat některý z úkonů, musí mít jednu z rolí, která takové právo obsahuje.

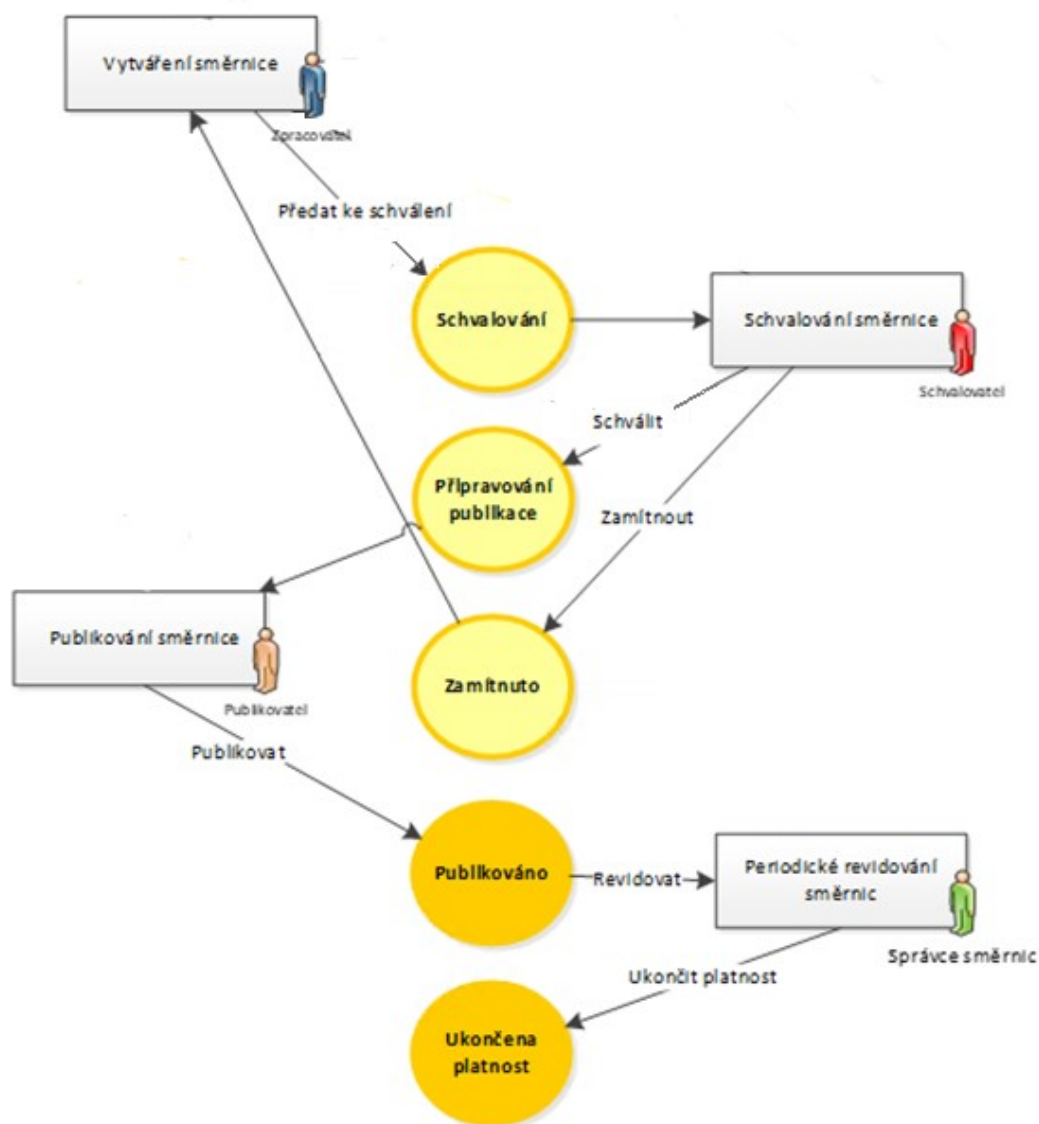
Agent	Název	Nutná aplikační podmínka	Guideline Processor	Guideline Commentator	Guideline Approver	Guideline Publisher	Guideline GuidelineManager	TaskGuideline WrittenBy	TaskGuideline AssignedTo
GUIDELINE	Publikovat směrnici bez schválení	PrimaryObject.State==...							
GUIDELINE	Revidování směrnice	SecurityObject.Validity...							X
GUIDELINE	Stomování potvrzení seznámení (b...	PrimaryObject.Confirm...							
GUIDELINE	Stomovat potvrzení seznámení	PrimaryObject.Confirm...							
GUIDELINE	Ukončení revidování směrnice (s ú...	SecurityObject.Validity...							X
GUIDELINE	Upravit záznam Směrnice		V			V	V		
GUIDELINE	Vložit záznam Určeno pro	SecurityObject.State !...	X			X	X		
GUIDELINE	Upravit záznam Určeno pro	SecurityObject.State !...	X			X	X		
GUIDELINE	Vrácení k potvrzení seznámení asi...	PrimaryObject.Confirm...							
GUIDELINE	Vrátit směrnici k doplnění (bez úkolu)	PrimaryObject.State==...				X			
GUIDELINE	Přidat záznam Souvisejících směrnic								

Obrázek 6: RightsViewer - základní role a jejich práva k vybraným úkonům

4.3 Modul Směrnice

Ze začátku praxe jsem se věnovala především agendě týkající se směrnic, na které jsem si osvojila specifické konstrukce v kódu a vůbec způsob, jakým je aplikace navržena a programována. Pokusím se tedy na ní podrobněji popsat základní principy fungování aplikace i z hlediska způsobu implementace. Úkoly v dalších agendách pak zmíním již spíše okrajově, bez širšího kontextu.

Modul Směrnice se zabývá evidencí směrnic a dohledem nad jejich životním cyklem. Každá směrnice od svého vytvoření postupně prochází mnoha stavy a na těchto změnách se může podílet větší množství uživatelů. Na obrázku 7 je výřez diagramu popisující tyto stavy a úkony, které celý proces doprovázejí. U jednotlivých uživatelů lze také vidět, jakou roli musí mít, aby mohli směrnici zpracovávat a přesouvat do dalších stavů.

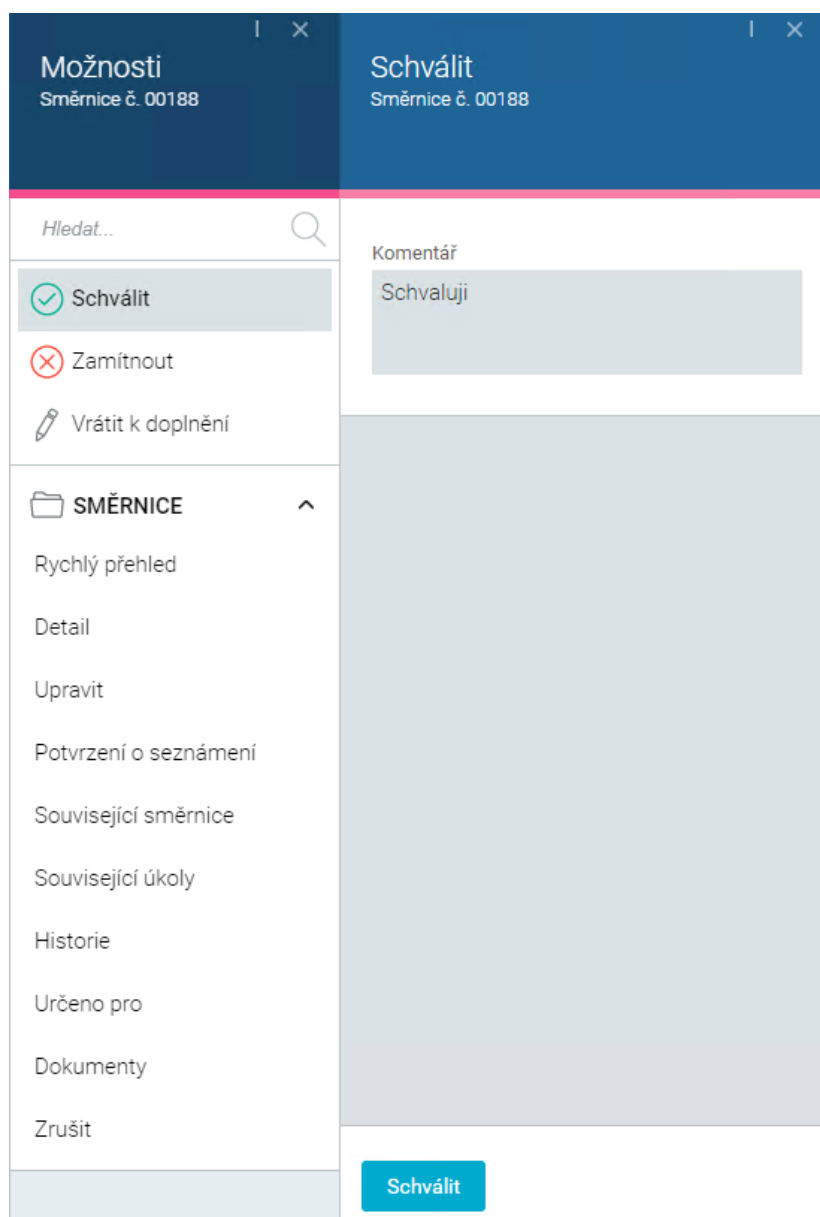


Obrázek 7: Schvalovací proces směrnice – přehled stavů

Jak již bylo řečeno, po kliknutí na konkrétní směrnici ze seznamu se zobrazuje panel *Možnosti*, kde je pro každou směrnici několik tlačítek s různými úkony, které se dají s tímto záznamem provádět. Mezi mé první úkoly patřilo přidávání dalších takových možností.

Pro dokončení celého schvalovacího procesu bylo nutné přidat několik úkonů, které posouvaly stav směrnice. Poté, co byla směrnice vytvořena, se dostala do stavu *Schvalování* a úkolem pověřené osoby, schvalovatele, bylo tuto směrnici schválit, zamítnout, či vrátit k doplnění. Bylo mi zadáno zapracovat do aplikace možnost *Schválit*.

Jednalo se konkrétně o přidání dalšího tlačítka mezi úkony, které po stisknutí otvíralo nový panel (viz obrázek 8). Ten obsahoval pouze textbox pro komentář a tlačítko *Schválit*, které pak provádělo schválení směrnice, tedy posun do dalšího stavu – *Připravování publikace*.



Obrázek 8: Panel Schválit

Po dosažení schválení pak v procesu následovaly další volby – *Vrátit k doplnění* nebo *Publikovat* (přesun do dalšího stavu). Musela jsem opět vytvořit tlačítko a panel pro tuto možnost (viz obrázek 9). Obsahoval výběr ze dvou radio buttonů, povinný textbox pro *Publikovaný dokument* s vyhledáváním a rozbalovacím menu s přidruženými dokumenty k dané směrnici a zase i obyčejný textbox a tlačítko pro dokončení úkonu.

The image shows a software interface for publishing a directive. The main panel is titled 'Publikovat' and 'Směrnice č. 00188'. It includes a search bar 'Hledat...', a 'Vrátit k doplnění' button, and a 'Publikovat' button. A sidebar on the left lists navigation options under 'SMĚRNICE'. The main area contains radio buttons for 'Publikovat směrnici' (selected) and 'Publikovat formální změnu'. Below these is a dropdown menu for 'Publikovaný dokument' with the selected item 'Kupní smlouva garáže.docx (0.1.0)'. A section for 'Shrnutí obsahu směrnice' is present but empty. At the bottom right is a blue 'Publikovat' button.

Obrázek 9: Panel Publikovat

Později jsem přidávala obsahově totožný panel i ke stavu *Schvalování*, pouze se změnou kontextu. V daném případě šlo totiž o *Publikovat bez schválení*.

Když se již směrnice nacházela ve stavu *Publikováno*, mělo být možné ji průběžně kontrolovat a revidovat. Pro každou směrnici se totiž při jejím vytváření musí vyplnit i termín ukončení platnosti a doba, kdy se již může zkontrolovat a rozhodnout o prodloužení či ukončení platnosti, případně požádat o revizi. Ve chvíli, kdy se směrnice dostává do období, kdy je již potřebná kontrola, přibude odpovědné osobě v panelu *Možnosti* u této směrnice volba pro její zkontrolování.

Na obrázku 10 lze vidět tlačítko a panel *Kontrola platnosti směrnice*, který jsem vytvořila pro tuto událost. Jeho obsahem byly tři radio butony s výše zmíněnými volbami, pole *Zpracovatel* s rozbalovacím menu, pole *Vyřešit do* s kalendářem pro datum a textbox. Následující dění pak vycházelo z vybrané volby. Pokud šlo o prodloužení platnosti, změny se pouze uložily k příslušnému záznamu. Při výběru možnosti *Ukončit platnost* se nejdříve jenom měnil stav směrnice v databázi. Později jsem tuto volbu rozšiřovala o další funkčnost, a to stornování souvisejících úkolů a záznamů se zrušenou směrnicí, které po jejím zneplatnění postrádaly smysl. V případě žádosti o revizi šlo o poměrně komplikovanější proces, který byl spojen s vygenerováním úkolu *Proved'te revizi směrnice*, který musel být splněn přiřazenou osobou. (podrobněji o úkolech v kapitole 4.4 Modul Úkoly).

Na obrázku níže lze vidět, že některá pole mohou být určena jako povinná a je třeba je ověřit. V tomto případě to je pole *Zpracovatel* a *Vyřešit do*. Validace probíhala jak na straně klientské, tak na straně serveru. Muselo se zkontrolovat vyplnění polí a jejich smysluplnost.

Možnosti

Směrnice č. 00063

Kontrola platnosti směrnice

Směrnice č. 00063

Hledat...

Zahájit přípravování

Kontrola platnosti ...

Revidovat

Revidováno

SMĚRNICE

Rychlý přehled

Detail

Upravit

Potvrzení o seznámení

Související směrnice

Související úkoly

Historie

Určeno pro

Dokumenty

Ukončit platnost

Prodloužit platnost

Požádat o revizi

Zpracovatel *

Opletalová Aneta

Vyřešit do *

01.05.2019 12:00

Komentář

Zkontrolujte platnost této směrnice

Uložit

Obrázek 10: Panel Kontrola platnosti směrnice

Na následujících úryvcích zdrojového kódu je ukázka implementace tohoto příkladu na klientské části. Na podobném principu byly vytvořeny i předchozí nové prvky.

Základem je určení vztahů mezi panely a daty, která se mezi nimi budou přenášet. Ve zdrojovém kódu 1 lze tuto relaci vidět - vztah mezi panelem *Možnosti směrnice* (*Guideline Options*) a panelem *Kontrola platnosti směrnice* (*Guideline Check Validity*), které si mezi sebou předávají identifikační číslo směrnice (*Connector GuidelineId*), která je aktuálně zpracovávána.

```
<Relation Key="GuidelineOptions.GuidelineCheckValidity.GuidelineId"
          WorkItem="GuidelineCheckValidity" InConnector="GuidelineId">
  <Parameters>
    <Parameter Key="OutWorkItem" Value="GuidelineOptions" ValueMode="WorkItem" />
    <Parameter Key="OutConnector" Value="GuidelineId" ValueMode="Object" />
    <Parameter Key="ConnectOnlyToParent" Value="true" ValueMode="Object" />
  </Parameters>
</Relation>
```

Zdrojový kód 1: Relace mezi panely

Celý proces zpracovávání dat a jejich zaslání na server je ovládán množstvím speciálních funkcí tzv. *extenzí*. Zdrojový kód 2 obsahuje jednu *extenzi*, která se stará o přenos *Connectoru* (identifikačního čísla směrnice) na server. Je zde specifikován název, pod kterým se získaná hodnota bude na serveru zpracovávat – *Key*, a také kdy má dojít ke spuštění *extenze* – při odeslání celého formuláře na server a kliknutí tlačítka *Uložit* (*Submit*).

```
<Extension Type="SetInConnectorValue2ControlDataPropertyWIEExtension">
  <Parameters>
    <Parameter Key="ControlData" ValueMode="ControlData" Value="FormControlData" />
    <Parameter Key="InConnector" Value="GuidelineId" ValueMode="InConnector"/>
    <Parameter Key="PropertyName" Value="Constants.Key" ValueMode="StaticModel" />
  </Parameters>
  <Triggers>
    <ControlDataEventTrigger ControlData="FormControlData" Event="BeforeCommand"/>
    <UIComponentEventTrigger UIComponent="\View\Panel\Form" Event="Submit"/>
  </Triggers>
</Extension>
```

Zdrojový kód 2: Extenze pro přenos dat na server

Nyní ukázka ze serverové části, na které nejdříve probíhá validace zaslaných záznamů. Zdrojový kód 3 zobrazuje ověření vstupů při výběru možnosti *Požádat o revizi*. Musí se zkontrolovat, zda byla vyplněna povinná pole *Zpracovatel* a *Vyřešit do*. Tyto hodnoty jsou z klientské části zaslány pod klíči `GUIDELINES.CREATED_BY_IDX` a `GUIDELINES.SOLVE_UNTIL`.

```
if (data.GetValue<string>(CheckValidityInputItem) == "RequestRevision")
{
    validationResult.CheckRequiredColumns(data, GUIDELINES.CREATED_BY_IDX,
                                           GUIDELINES.SOLVE_UNTIL);
}
```

Zdrojový kód 3: Validace povinných polí

Po kontrole se na základě hodnot získaných z formuláře vytváří úkol. Vždy se přiřazuje vazba na konkrétní směrnici, řešitel a termín splnění. Příklad vytvoření takového úkolu je zobrazen ve zdrojovém kódu 4.

```
MwrTaskEntity task = CreateEntity<MwrTaskEntity>();
task.PrimID = PrimaryEntity.Idx;
task.AssignedToID = data.SafeGetValue<Idx>(GUIDELINES.CREATED_BY_IDX);
task.Deadline = data.SafeGetValue<DateTime>(GUIDELINES.SOLVE_UNTIL);

CreateTask(task);
```

Zdrojový kód 4: Vygenerování úkolu

V panelu se kromě úkonů souvisejících se schvalovacím procesem nacházejí i další možnosti. Každá tato možnost pak otevírá opět další panely, ve kterých jsou informace a podrobnosti týkající se daného záznamu. Téměř vždy se jedná o *Detail*, *Rychlý přehled*, *Upravit* a *Související úkoly*. Pouze pro záznam z agendy Směrnice jsou pak přidány specifické přehledy zahrnující *Související směrnice*, *Seznam potvrzení o seznámení* a *Určeno pro*.

Já jsem upravovala pohled *Určeno pro* – seznam uživatelů (konkrétních osob nebo i v rámci rolí či organizačních jednotek), kterým je směrnice určena a je nutné, aby si ji přečetli a potvrdili jejich přečtení. Záznamy v tomto seznamu je opět možné upravit nebo odstranit. Doplněvala jsem ještě volbu *Odstranit*, takže opět tlačítko a panel pro dokončení úkonu. Po stisknutí tlačítka *Uložit* se tento záznam stane nevalidním a nebude již zobrazován v seznamu. Celý tento pohled lze vidět na obrázku 11.

Poslední možnost, kterou jsem přidávala, byla evidence souvisejících úkolů (podrobněji o úkolech v kapitole 4.4 Modul Úkoly).

Možnosti

Směrnice č. 00154

Uřčeno pro

Směrnice č. 00154

+

↺

Nový

Obnovit

Možnosti

Firma A, s.r.o. - Opletalová Anežka

Odstranit

Firma A, s.r.o. - Opletalová Anežka

Hledat...

Vyžádat připomínky

Předat ke schválení

Publikovat bez sch...

SMĚRNICE

Rychlý přehled

Detail

Upravit

Potvrzení o seznámení

Související směrnice

Související úkoly

Historie

Uřčeno pro

Dokumenty

Zrušit

Hledat...

URČENO PRO

FIRMA

Vedoucí marketingu

Firma A, s.r.o.

Role ve firmě

100 - Výrobní oddělení

Firma A, s.r.o.

Organizační jednotka

Opletalová Anežka

Firma A, s.r.o.

Pracovník

Asistentka

Firma A, s.r.o.

Role v organizační jednotce

Oddělení sekretářských služeb - Od...

Hledat...

DALŠÍ

Upravit

Odstranit

Opravdu chcete odstranit záznam ze seznamu Uřčeno pro?

Odstranit

Obrázek 11: Panel Uřčeno pro a Odstranit záznam

Výše jsem popisovala, jak se směrnice musí pravidelně kontrolovat a revidovat. Uživatel by tak sice viděl u směrnice s blížícím se datem skončení platnosti možnost *Kontrola platnosti*, ale je nesmyslné všechny směrnice pravidelně procházet a sledovat, zda se tlačítko již objevilo.

Tyto termíny by měla hlídat aplikace a včas na ně upozornit. Musela jsem implementovat tzv. *actionflow*, které se bude pravidelně automaticky spouštět a provádět kontrolu nad všemi směrnicemi. V případě myTeamu se tyto periodické akce vykonávají vždy v jednu hodinu v noci. Pro tuto funkcionalitu jsem nejprve vytvořila metodu, která postupně projde všechny směrnice a uloží si je do seznamu. Dalším krokem (viz Zdrojový kód 5) je pak v tomto seznamu najít takové záznamy, u kterých se blíží termín ukončení platnosti. Neeviduje se však konkrétní datum, kdy začíná období pro kontrolu, ale pouze počet měsíců (popř. dnů), které délku tohoto období určují. Přesné datum se proto musí vypočítat odečtením daného počtu měsíců od aktuálního data.

Když se za těchto podmínek nějakou směrnicí podaří najít, je potřeba upozornit kompetentní osobu, aby se postarala o kontrolu. Z tohoto důvodu se v reakci na tento požadavek vygeneruje úkol *Zkontrolujte platnost směrnice* (po ověření, zda už takový úkol neexistuje) a jako řešitel je určena odpovědná osoba.

```
List<Id> guidelines = GetGuidelines();
foreach (Id guidelineItem in guidelines)
{
    GuidelineEntity guidelineEntity = GetEntityByKey<GuidelineEntity>(guidelineItem);

    reminderDate = guidelineValidTo.AddMonths(guidelineEntity.monthsToRemind * (-1));

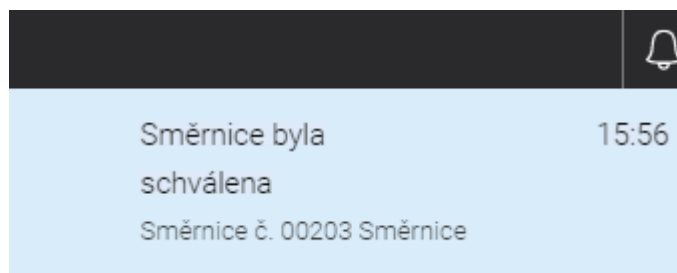
    if (DateTime.Today >= reminderDate)
    {
        TaskEntity guidelineTask = GetTaskEntity(guidelineEntity.Idx);

        if (guidelineTask == null)
        {
            GenerateTaskGuidelineValidity(guidelineEntity);
        }
    }
}
```

Zdrojový kód 5: Periodická kontrola směrnice

Zmíněné procesy a změny stavů doprovází také průběžné upozorňování zúčastněných osob. Na obrázku 12 je jednoduchá notifikace určena právě pro tyto účely. Uživatelé se zobrazují v pravém horním rohu aplikace pod ikonou zvonečku.

Takovou notifikaci jsem připravila například pro schválení směrnice. Když tedy schvalovatel provedl úkon *Schválit směrnici*, všem osobám, které o tom měly být informovány, vyskočilo pop up okno se zprávou.



Obrázek 11: Zvoneček s notifikací

Kromě upozornění přímo v aplikaci se také při různých událostech generují na uživatele e-maily. Může se opět jednat o nějakou změnu stavu, na kterou je zainteresovaná osoba pouze upozorněna, nebo nově vytvořený úkol, na který je upozorněn e-mailem jeho řešitel.

Tuto funkcionalitu jsem přidávala u stejného případu jako výše zmíněnou notifikaci, a to po schválení směrnice, kdy osobě, která směrnici vytváří a zpracovává, přijde s touto informací e-mail.

4.4 Modul Úkoly

Úkoly jsou základem celé aplikace a jsou generovány různými procesy a změnami stavů v rámci všech agend. U úkolu se zaznamenává kromě jeho obsahu a řešitele také například zadavatel (může být i samotný informační systém), termín splnění nebo jeho aktuální stav, který se postupně ukládá i do samostatné entity s historií úkolu. V modulu s úkoly se nachází seznam všech úkolů, které se aktuálně řeší nebo jsou už vyřešeny (viz obrázek 13). Jejich evidenci tedy pokrývá samostatná agenda, ale jejich splnění je možné vždy i z agendy, ke které se úkol vztahuje – např. Směrnice.

Pokud ale vyberu úkol, který je přidružený nějaké směrnici (např. *Schvalte směrnici*, *Zkontrolujte platnost směrnice*), zobrazí se v panelu možností nejen úkony a podrobné informace k úkolu, ale také základní nabídka konkrétní směrnice, ke které úkol patří. Tuto podskupinu možností jsem doplňovala nejen pro směrnice, ale i pro některé další agendy. V tomto případě stačilo vytvořit pouze nový úsek s volbami, který se bude zobrazovat pouze u úkolů přiřazeným směrnici. Každá taková volba odkazuje na už vytvořené panely, ke kterým se dá dostat i přes záznam směrnice v jejím modulu.

Úkoly

+

↺

↻

📄

Nový

Odevozt

Filtery

Skupiny

Hledat...

Má se zabývat

STAV

PŘEDMĚT

VYŘEŠIT DO

ŘEŠITEL

PRIORITA

Hledat...

🔍

Kontrola platnosti ...

📄

ÚKOL

Ukončit platnost

Prodloužit platnost

Požádat o revizi

Zpracovatel

Opětaloová Aneta

Vyřešit do

30.03.2019 0:00

Komentář

➔	Zkontrolovat platnost - Směrnice Proces směrnice 00051	28.11.2018 st 9:56	Opětaloová Aneta	▼	<div>📄</div> <div>ÚKOL</div> <div>Rychlý přehled</div>	
➔	Zkontrolovat platnost - Směrnice Proces směrnice 00052	28.11.2018 st 9:56	Opětaloová Aneta	▼	<div>Detail</div> <div>Průběh řešení</div> <div>Dokumenty</div>	
➔	Zkontrolovat platnost - Směrnice Proces směrnice 00055	28.11.2018 st 9:56	Opětaloová Aneta	▼	<div>📄</div> <div>SMĚRNICE</div> <div>Rychlý přehled</div>	
➔	Zkontrolovat platnost - Směrnice Proces směrnice 00056	28.11.2018 st 9:56	Opětaloová Aneta	▼	<div>Detail</div> <div>Dokumenty</div>	
➔	Zkontrolovat platnost - Směrnice Proces směrnice 00057	28.11.2018 st 9:56	Opětaloová Aneta	▼	<div>Detail</div> <div>Dokumenty</div>	
➔	Zkontrolovat platnost - Směrnice Proces směrnice 00055	28.11.2018 st 9:56	Opětaloová Aneta	▼	<div>Detail</div> <div>Dokumenty</div>	
➔	Zkontrolovat platnost - Směrnice testrevize Proces směrnice 00062	28.11.2018 st 9:57	Opětaloová Aneta	▼		
➔	Zkontrolovat platnost - Směrnice Proces směrnice 00063	28.11.2018 st 9:57	Opětaloová Aneta	▼		
➔	Potvrdit seznámení - Směrnice Test schvalování	07.12.2018 st	Opětaloová Aneta	▼		<div>Uložit</div>

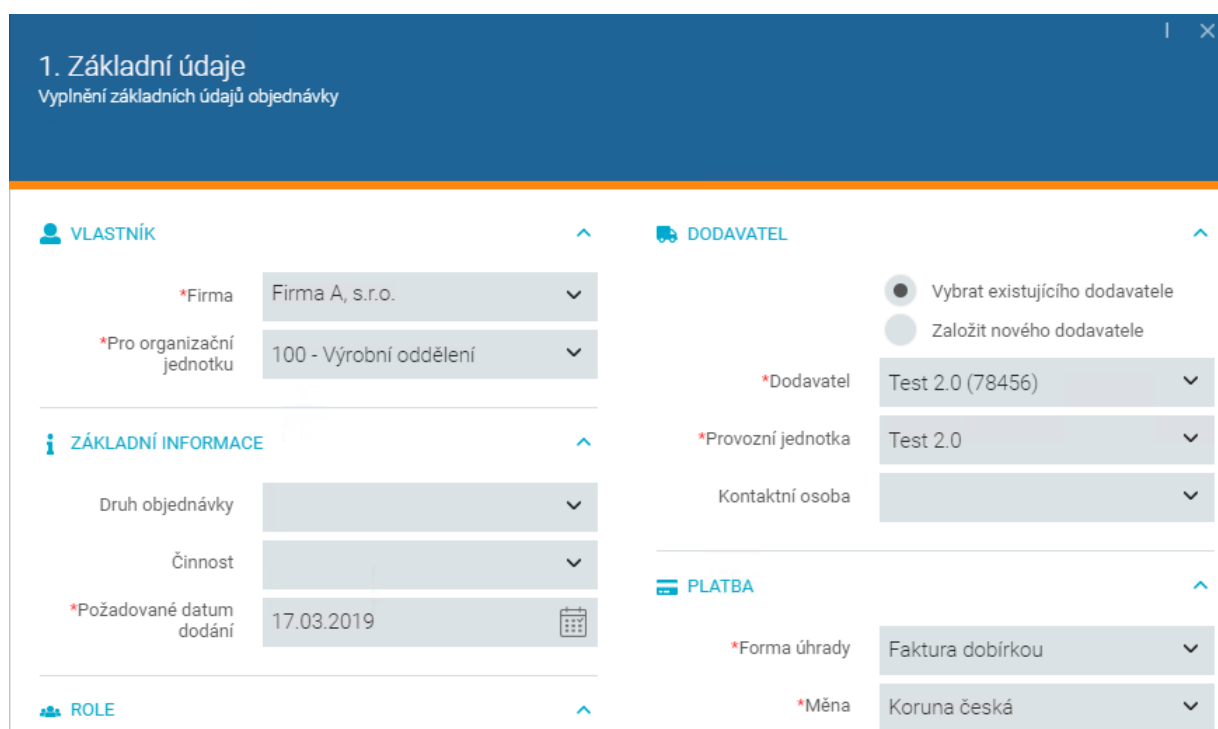
Obrázek 13: Modul Úkoly s pohledem pro směrnice

4.5 Modul Objednávky

Později jsem se začala věnovat práci na dalších agendách, kde většinou již nešlo o tvorbu nových panelů, ale spíš o vylepšování funkcionality. Jednou z nich byla agenda objednávek.

Nejdříve jsem upravovala panel pro *Nový záznam*, který lze vidět na obrázku 14. Jednalo se vytvoření a zaevidování nové objednávky. Mým úkolem bylo přidat k tomuto panelu pole pro *Provozní jednotku* a následně na základě této hodnoty automaticky vybrat výchozí hodnotu pro pole *Forma úhrady*. Vytvořila jsem nový povinný textbox s vyhledáváním a rozbalovacím menu s přidruženými záznamy provozních jednotek daného dodavatele. Toto pole tedy zůstávalo needitovatelné, dokud uživatel nevyplnil pole *Dodavatel*.

Požadavkem také bylo ihned po vyplnění dodavatele nastavit v poli *Provozní jednotka* výchozí hodnotu (pokud existuje), a tou byla dodavatelova jednotka s příznakem „Centrála“. Po automatickém dosazení výchozí hodnoty, nebo manuální změně uživatelem se ihned i do pole *Forma úhrady* měla přednastavit výchozí hodnota pro konkrétní provozní jednotku. Jestliže ale žádná výchozí hodnota pro formu úhrady nebyla u záznamu nalezena, pak se musela dosadit ze záznamu dodavatele. Podobnou funkčnost jsem poté aplikovala i u dalších agend.



Obrázek 12: Objednávky vydané – Panel Nový záznam

Při vytváření textboxu se na klientské straně neurčuje pouze jeho vzhled, ale také požadovaná funkčnost. Jsou to například podmínky, za jakých je povolena editace, zda je zpřístupněno fulltextové vyhledávání nebo jestli se musí pole povinně vyplnit. Nutností také je určit a pojmenovat data, která se budou zpracovávat a následně zobrazovat v textboxu nebo rozbalovacím menu a odkud je získat.

O přenos dat se opět starají *extenze*. Zdrojový kód 6 definuje některé parametry textového pole – označení klíčů, pod kterými budou tato data vystupovat a název serverové třídy, která dodá záznamy pro rozbalovací menu.

```
<Parameter Key="DataSourcePropertyName" Value="OperatingUnit._IDX.FullName, "  
                                "OperatingUnit.NAZEV.FullName"/>  
<Parameter Key="ControlData" ValueMode="ControlData" Value="OperatingUnitData" />
```

Zdrojový kód 6: Textbox Provozní jednotka

Zdrojový kód 7 obsahuje úryvek jedné metody ze třídy *OperatingUnitData* (viz Zdrojový kód 6), ve které se sestavuje SQL dotaz získávající hodnoty pro již zmiňované rozbalovací menu. Jde o spojení tabulky *Provozní jednotka* a *Dodavatel*, přičemž jsou dohledány záznamy o provozních jednotkách přidružené konkrétnímu dodavateli, který je již vybrán uživatelem.

```
SelectQuery query = new SelectQuery();  
  
query.From.BaseTable = OperatingUnit;  
  
query.From.AddInnerJoin(Partner, SqlExpression.And(  
    OperatingUnit.Table.PARTNER_IDX.Eq(Partner.Table._IDX)));  
  
query.Select.Add(OperatingUnit.Table.IDX, OperatingUnit.Table.NAZEV);  
  
return query;
```

Zdrojový kód 7: Skládání SQL dotazu pro rozbalovací menu Provozní jednotky

Po přípravě samotného textového pole se tedy dostávám k vyplňování výchozí hodnoty *Forma úhrady*. Při zpracovávání tohoto požadavku se na serveru nejdříve musí rozhodnout, jaká data jsou vyplněna v textbozech. V extenzi, jejíž část je zobrazena ve zdrojovém kódu 8, jsem určila, že se data budou přenášet až v reakci na změnu pole *Dodavatel* nebo *Provozní jednotka*.

```
<ComponentEventTrigger Component="\Panel\Form\Partner" Event="ValueChanged"/>  
<ComponentEventTrigger Component="\Panel\Form\OperatingUnit" Event="ValueChanged"/>
```

Zdrojový kód 8: Triggery pro spuštění extenze

Pokud je vyplněn *Dodavatel* i *Provozní jednotka*, zavolá se metoda, která pomocí SQL dotazu vrací formu úhrady přiřazenou u záznamu s provozní jednotkou. V opačném případě, kdy známe jen dodavatele, vrací další metoda jeho výchozí formu úhrady. Tuto funkcionalitu zajišťuje kód ve zdrojovém kódu 9.

```
SelectQuery query = new SelectQuery();

if (Parameters.SecondaryKeys.TryGetValue(OperatingUnit.IDX.FullName)
{
    query = GetPaymentTypeForOperatingUnit(operatingUnitIdx);
}
else
{
    query = GetPaymentTypeForPartner(partnerIdx);
}

return query;
```

Zdrojový kód 9: Přednastavení pole Provozní jednotka – část 1

Může se ovšem stát, že záznam o provozní jednotce v databázi bude neúplný a nenajdou se potřebná data. Tuto možnost jsem musela ověřit (viz Zdrojový kód 10) a pokud nastala, opět se provedl pokus o získání výchozí hodnoty formy úhrady, tentokrát ale ze záznamu o dodavateli. Nově získaná data se pod svými klíči odesílají na klienta.

```
string paymentType = dataRecord.SafeGetValue<string>(PaymentType.Table.IDX).Trim();

if (string.IsNullOrEmpty(paymentType))
{
    DataRow result = GetPaymentTypeForPartner(partnerIdx);
    dataBagObject.Add(UHRADA_FORMA.IDX,
        result.GetValue<Idx>(PaymentType.Table._IDX));
}
```

Zdrojový kód 10: Přednastavení pole Provozní jednotka – část 2

4.6 Modul Podatelna

Poslední modul, který bych chtěla ještě zmínit, je Podatelna. V této agendě se evidují přijaté a odeslané zásilky.

Dle obrázku 15 lze poznat, že se opět jednalo o rozšíření panelu pro nový záznam, ovšem tentokrát pro odeslanou zásilku. Mezi stávající způsoby odeslání zásilky jsem přidala variantu *Dopis Online*. Sekce pro *Dopis Online* se měla zobrazovat jen ve chvíli, kdy si uživatel vybere z nabídky v poli *Způsob odeslání*, že právě tento typ zásilky chce vytvořit. Poté se tedy rozbálí, do této chvíle skrytý, úsek pro doplnění údajů k zásilce typu *Dopis Online*. V sekci se nachází dvě zaškrtačací tlačítka, dvě pole s rozbalovacím menu, z nichž *Kategorie zásilky* je povinná a po otevření sekce již obsahuje výchozí hodnotu „Obyčejně“.

Po zkompletování zásilky a stisknutí tlačítka *Uložit* se záznam ukládá do databáze.

Obrázek 13: Odeslaná zásilka – Panel Nový záznam

K tomuto příkladu jsem vybrala ukázkou podmínky na klientské straně, která ověřuje, zda je *Způsob odeslání* právě *Dopis Online* (hodnota 40). Porovnává se hodnota parametru *ShipmentChannel*, v němž je uložen způsob odeslání vybraný v příslušném textovém poli. Pokud se nejedná o *Dopis Online*, úsek má vlastnost *Disabled*. Tuto podmínku znázorňuje zdrojový kód 11.

```
<Parameter Key="Disabled"
  Value="(@{ControlData}.getData()[@{PropertyNameShipmentChannel}] != 40)"
  IsExpression="true">
  <Parameters>
    <Parameter Key="ControlData" Value="FormControlData"
      ValueMode="ControlData"/>
    <Parameter Key="PropertyNameShipmentChannel"
      Value="SHIPMENTS_OUT.SHIPMENT_CHANNEL.FullName" />
  </Parameters>
</Parameter>
```

Zdrojový kód 11: Sekce Dopis Online – podmínka zobrazení

4.7 Sjednocení e-mailu

Jedním z mých časově rozsáhlejších úkolů bylo sjednocení vzoru pro veškeré e-maily, které se odesílaly ze všech agend napříč celou aplikací. Dosavadně byly posílány všechny v jiné vizuální formě, vycházely z mnoha různých *templatů* a zbytečně se pro každý nově posílaný e-mail vytvářel *template* nový, který odpovídal pouze aktuálnímu požadavku na vzhled a obsah informací. Po prověření aktuálního stavu jsem se dostala asi k 20 odlišným způsobům, jakými se vytvářelo tělo e-mailu.

Nejdříve jsem si tedy po konzultaci s analytiky a grafikem navrhla tři vzory pro různé typy e-mailu. Jeden byl čistě jednoduchý text, další měl být rozesílán při změnách stavu (např. při reakci na *Schválení směrnice*) a poslední se generoval řešitelům na základě nově přiřazených úkolů ke splnění (např. úkol *Zkontrolujte platnost směrnice*).

První případ byl samozřejmě nejjednodušší, šlo pouze o přepsání původních velmi podobných tříd na třídu novou, která se od nich téměř neodlišovala a jen je sjednocovala. Další *templaty* byly poměrně složité nejen na vytvoření, ale také na jejich zpracování do aplikace a nahrazení těch prvotních. Musela jsem nejdříve projít podrobně všechny stávající vzory a sepsat si, co každý z nich obsahoval ve svém těle. Bylo nutné určit, které základní údaje se budou vyskytovat v každém e-mailu a které jsou specifické pro konkrétní případ, a jak je zpracovat.

Výsledkem byl vzor, který se skládal z hlavičky, předmětu a těla. Hlavička obsahovala logo myTeamu, odkaz na konkrétní položku a datum změny s krátkým popisem. Předmětem byl nadpis, který objasňoval, čeho se e-mail týká. Tělo obsahovalo zprávu, která byla seskládána z různých informací a dat příslušících konkrétnímu použití. Případně ještě byla přiložena poznámka od předchozích zpracovatelů. Vše jsem pak nastylovala pomocí HTML a CSS. Když jsem tedy měla vytvořeny tyto tři vzory, musela jsem je zpracovat namísto těch původních. Postupně jsem měnila jejich strukturu, aby všechny názvy a data odpovídaly těm, které jsem potřebovala nově zahrnout do těla e-mailu. Příklad dvou e-mailů lze vidět na obrázcích 15 a 16.

myTEAM®

Objednávka č. [19-A100-00088](#)

Ze dne 06.03.2019 0:00:00

Zamítnutí objednávky

Objednávka **Dell CZ (546,00 Kč)** byla vrácená k doplnění!

Schvalovatel 2 Test

Komentář: vraceno

Obrázek 14: E-mail Zamítnutí objednávky

myTEAM®

[Porada](#)

Doba trvání 01.23.2019 05:00 - 01.23.2019 05:30

Pozvánka na poradu

Zhodnocení roku

Přijměte, prosím, pozvánku na poradu „**Zhodnocení roku**“, která se bude konat **23.01.2019, 17:00 - 17:30** hodin.

Místo konání: VENTUS

Obrázek 15: E-mail Pozvánka na poradu

4.8 Issues

Nedílnou součástí vývoje softwaru je také opravování chyb, které se projeví v průběhu testování nebo používání aplikace. Pokud se objevila zásadní chyba u klienta, která vyžadovala okamžitou opravu, věnovali jsme se tzv. *issues*. *Issue* je tedy chyba v aplikaci, kterou vypisuje tester, když na ni narazí při vlastním testování, nebo na základě zpětné vazby od klienta.

5 Závěr

Prací ve firmě Kvados, a. s. jsem strávila mnohem více než stanovených 50 dní a jsem velmi ráda, že jsem příležitost absolvovat individuální odbornou praxi využila. Mohla jsem na reálném projektu využít své dosavadní znalosti a díky dalším školením, které mi byly zprostředkovány, se obohatit o nové. Naučila jsem se používat nástroje pro týmový vývoj a programovat na základě analýz. S přibývajícimi znalostmi jsem se pak mohla čím dál tím více podílet na vývoji aplikace myTeam, což bylo samozřejmě mým cílem.

Po celou dobu praxe jsem ale usilovala nejen o zlepšení svých programátorských dovedností, ale také o nahlédnutí do struktury firmy, která se věnuje informačním technologiím. Obrovským přínosem pro mě tedy bylo i samotné začlenění do vývojového týmu. Dozvěděla jsem se, jak probíhá vývoj softwaru a kolik lidí napříč různými pozicemi se na výsledném produktu podílí. Osvojila jsem si praktiky scrum metodologie a porozuměla jsem firemním procesům, které doprovázely mé působení ve firmě.

Po skončení praxe mi byla nabídnuta i následná spolupráce, a proto věřím, že má účast na projektu myTeam byla přínosná pro obě strany. Svou praxi tedy hodnotím velmi kladně.

Literatura

- [1] KVADOS, a. s.. *KVADOS, a. s.* [online]. Copyright © KVADOS, a. s. Všechna práva na obsah vyhrazena. Loga a jména produktů jsou ochrannými nebo registrovanými ochrannými známkami akciové společnosti KVADOS a jiných držitelů. [cit. 24.03.2019]. Dostupné z: <https://www.kvados.cz/>
- [2] myTEAM® - systém pro řízení vnitrofiremních procesů, dokumentů a sdílení informací | KVADOS, a. s.. *KVADOS, a. s.* [online]. Copyright © KVADOS, a. s. Všechna práva na obsah vyhrazena. Loga a jména produktů jsou ochrannými nebo registrovanými ochrannými známkami akciové společnosti KVADOS a jiných držitelů. [cit. 24.03.2019]. Dostupné z: <https://www.kvados.cz/produkty/myteam/>
- [3] *"Application Lifecycle Management with Visual Studio and Team Foundation Server". MSDN. Microsoft. 2013. Retrieved 2013-10-15.*
- [4] Sdílejte kód. Sledujte práci. Dodávejte software. | Team Foundation Server - Visual Studio. *Visual Studio IDE, Code Editor, Azure DevOps, & App Center - Visual Studio* [online]. Dostupné z: <https://visualstudio.microsoft.com/cs/tfs/>
- [5] Oficiální domovská stránka Microsoft. *Microsoft Corporation* [online]. Dostupné z: <https://www.microsoft.com/cs-cz>
- [6] Visual Studio IDE, Code Editor, Azure DevOps a App Center - Visual Studio. *Visual Studio IDE, Code Editor, Azure DevOps, & App Center - Visual Studio* [online]. Dostupné z: <https://visualstudio.microsoft.com/cs/>
- [7] Průvodce jazykem C# | Microsoft Docs. [online]. Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/csharp/>
- [8] *AngularJS Official Site* [online]. [cit. 2018-05-18]. Dostupné z: <https://angular.io/>
- [9] SQL Server Management Studio (SSMS) - SQL Server | Microsoft Docs. [online]. Dostupné z: <https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-2017>
- [10] SQL Server 2017 v systémech Windows a Linux | Microsoft. *Microsoft Corporation* [online]. Copyright © Microsoft 2019 [cit. 06.04.2019]. Dostupné z: <https://www.microsoft.com/cs-cz/sql-server/sql-server-2017>
- [11] Kanban Board Template for Agile PM. *Excel Templates, Calendars, Calculators and Spreadsheets* [online]. Copyright © 2003 [cit. 24.03.2019]. Dostupné z: <https://www.vertex42.com/ExcelTemplates/agile-kanban-board.html>
- [12] *Schwaber, Ken (February 1, 2004). Agile Project Management with Scrum. Microsoft Press. ISBN 978-0-7356-1993-7.*

- [13] SQL Server Profiler - SQL Server | Microsoft Docs. [online]. Dostupné z: <https://docs.microsoft.com/en-us/sql/tools/sql-server-profiler/sql-server-profiler?view=sql-server-2017>